RRM: Relightable assets using Radiance guided Material extraction

Diego Gomez[1][0009−0005−2847−8617], Julien Philip[2][0000−0003−3125−1614], Adrien Kaiser[2][0000−0002−5998−3932], and Élie Michel[2][0000−0002−2147−3427]

[1] École polytechnique
[2] Adobe Research

Abstract. Synthesizing NeRFs under arbitrary lighting has become a seminal problem in the last few years. Recent efforts tackle the problem via the extraction of physically-based parameters that can then be rendered under arbitrary lighting, but they are limited in the range of scenes they can handle, usually mishandling glossy scenes. We propose RRM, a method that can extract the materials, geometry, and environment lighting of a scene even in the presence of highly reflective objects. Our method consists of a physically-aware radiance field representation that informs physically-based parameters, and an expressive environment light structure based on a Laplacian Pyramid. We demonstrate that our contributions outperform the state-of-the-art on parameter retrieval tasks, leading to high-fidelity relighting and novel view synthesis on surfacic scenes.

Keywords: Image-based rendering· Reflectance modeling · Reconstruction · Computational photography· Machine learning

1 Introduction

The use of fully optimizable models as 3D scene representations to address novel view synthesis problems has led in the last years to impressive results: trained on a set of multiple photographs of a scene, these models can infer unseen view angles while using as sole prior the three-dimensionality of the underlying scene. Initially based on neural network overfitting (Neural Radiance Fields [15]), later approaches focused on improving the positional encoding fed as input to the networks (Fourier features [19], hash-grids of InstantNGP [16]), leading lately to neuron-free representations like TensoRF [3] (when used with Spherical Harmonics decoding) or 3D Gaussian Splatting [8].

The effectiveness of such overfit representation at retrieving 3D information even in the presence of transparent elements or strong specular effects makes it a strong competitor of traditional photogrammetry when it comes to acquiring 3D scenes from pictures. However, overfit representations usually encode only the radiance emitted by a scene, in a way that is hard to disentangle from their environment lighting at the time of acquisition. Hence a series of recent work focuses on relighting such overfit scenes, either by directly processing radiance data [20,

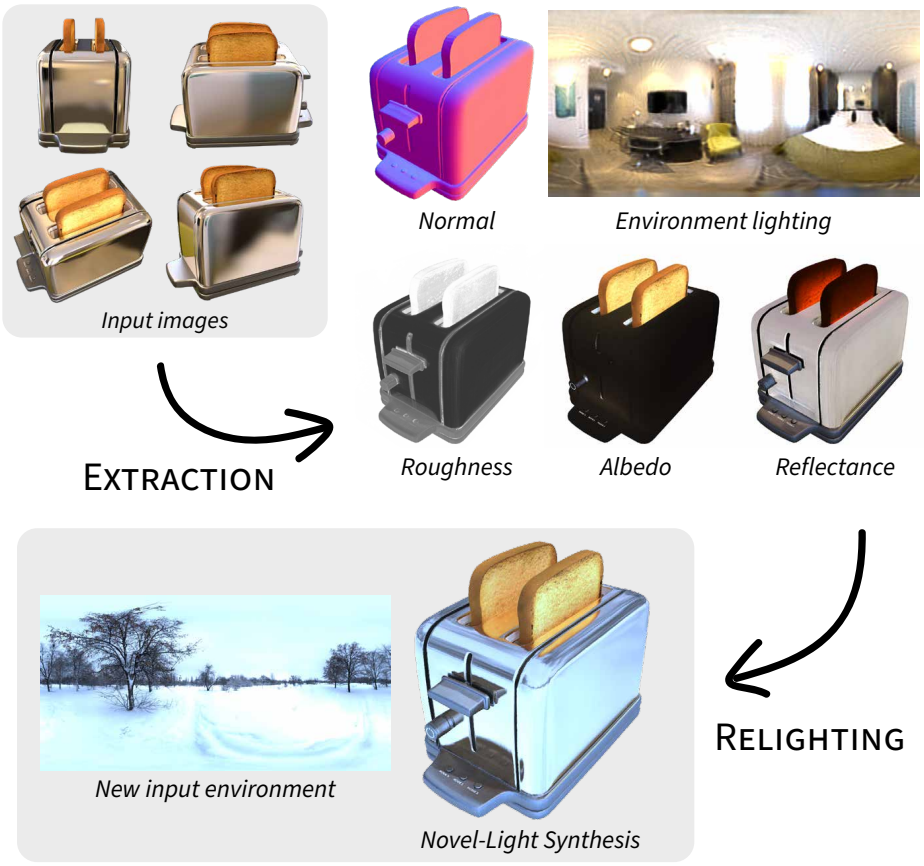

Fig. 1. We take as input a collection of photographs from a scene, and extract a model with physically-based parameters from which we can set a new lighting condition. In comparison to NMF [13] and TensoIR [7], our method is more robust to glossy materials and better handles self-reflection, as it is able to reconstruct more accurate surface normals.

24] or by extracting parameters compatible with physically-based 3D rendering pipelines [29, 7, 13]. Our work builds on the latter, improving the extraction capability of the model thanks to a more powerful representation of environment lighting. In particular, we better reconstruct the local **surface normal** from its appearance, even in the presence of highly glossy materials. Overall our key contributions are:

- The introduction of a physically aware radiance module that extracts coarse normals and a notion of roughness, while splitting the predicted radiance signal into view dependent and independent components.
- A novel way to represent environment maps based on a **Laplacian Pyramid** powered by a multiple importance sampling (MIS) algorithm that enables the retrieval of highly specular effects on complex geometry;
- A novel use of a radiance field as a guide to learning physically-based parameters. More specifically, a **supervision loss** on diffuse and glossy effects and the sharing of both explicit (normal, roughness) and underlying (appearance, 3D scalar fields) parameters allows disambiguating the incoming information, leading to the extraction of high-quality parameters.

2 Previous Work

Neural Scene Representation. Following their wide success in machine learning tasks, neural networks started being used as a means to encode high-dimensional data through overfitting. Typically applied to spatial fields (2D images, 3D signed distance fields, etc.), this approach has shown to be very good at compression and interpolation while providing fully differentiable random access look-up, hence being compatible with optimization tasks [17, 14]. It was thus a good fit to encode 5D radiance fields, whose storage had been a longstanding challenge of computer graphics [9, 4]. NeRF [15] demonstrated the use of neural network overfitting to optimize a volumetric representation whose (differentiable) render match predefined views and was soon followed by many similar approaches, progressively shifting the model's architecture towards positional encoding [16, 3].

A prominent challenge however lies in the user editing of such models. Recent efforts tackle the geometric transformations of neural representations or appearance editing of some kind [21, 6, 26]. A particularly important problem that has received attention in the past years is that of relighting [29, 18]. Some approaches attempt to tackle this by assuming known lights and parameterizing this information as an input of the model [20, 24]. Our work lies in the family of light-agnostic approaches, that involve leveraging inverse rendering to retrieve relightable assets [7, 13].

Neural Material Prediction. In the context of material generation, there exists precedent of predicting albedo, specular, normal and roughness parameters in order to render them into the desired result, for example GAN-based methods [31].

This is related to our physically-based module (see Fig. 2). Our physically-aware radiance module is then used to complement and inform the former. The radiance module inputs appearance related features and normals, to produce a radiance signal that we split into its view dependent and independent components.

Inverse Rendering. Inverse Rendering, the translation of observed images into global geometric, material, and lighting properties is a long-standing problem in computer vision and graphics. Being an extremely under-constrained problem it requires the introduction of several priors to achieve interesting results. These priors are typically provided by the structure of the differentiable renderer used to approach the task and the underlying scene representation. Some differentiable renderers are based on rasterization [5], point splatting [25], path tracing of globally illuminated meshes [1], ray marching through emissive volumes [15]. The usage of the differentiable renderer of choice dictates the priors that will be introduced to the system. In the case of NeRFs, the only prior is the 3-dimensionality of the scene, while in the case of rasterization one assumes a surfacic mesh. In our work two differentiable renderers are leveraged to perform the inverse rendering task. A physically-based one and a radiance-based one.

NeRFactor [29] and NeRV [18] present approaches that distill the information learned by a NeRF into a set of separate MLPs that predict geometric, visibility, and material information. These methods, however, have their limitations. NeRV requires the use of a dataset with known lighting conditions to incorporate indirect lighting information during training, whereas NeRFactor does not account for indirect lighting and self-reflections at all. TensoIR [7] greatly outperforms these previous methods in the task of inverse rendering. This is achieved by leveraging TensoRF [3] to replace the inaccurate prediction of the visibility parameter done by its predecessors. These methods, however, are not able to tackle scenes with specular objects.

PhySG [27] presents an inverse rendering pipeline that specializes in such objects. Nevertheless, this work does not take into account indirect illumination and thus fails at accounting for inter-reflection. The paper also restricts itself to using constant and monochrome specular BRDFs. Our method on the contrary handles both diffuse and glossy scenes, while enabling the modeling of inter-reflection and a BSDF that is spatially-varying on all components. This enables the retrieval of complex objects, such as the toaster in Fig. 1.

Neural Microfacet Fields (NMF) for Inverse Rendering [13] takes a similar approach to extend previous works to these challenging scenes. They do this by embedding in the 3D representation introduced by TensoRF [3] a microfacet representation. We however retrieve **higher quality parameters** thanks to our contributions which allow to better disentangle the incoming signal. The superior quality of the parameters we retrieve can be seen in the comparison we do in glossy scenes on the relighting task and our normal comparison quantitative results.

The recent work of NeRO [11] is able to faithfully reconstruct the geometry and the BRDF of real-life reflective objects. Their approach consists of **two stages**. First the geometry of the scene is retrieved with a neural SDF; then,

with the geometry fixed, an accurate BRDF of the object is computed. Our work in contrast consists of a pipeline that is **end-to-end optimizable**. Moreover, NeRO leverages radiance fields exclusively to learn the geometry of the scene. We show that these models are capable of providing much more than reliable geometry, indeed with the proper parameterization they can provide insightful information about physical properties.

3 Overview

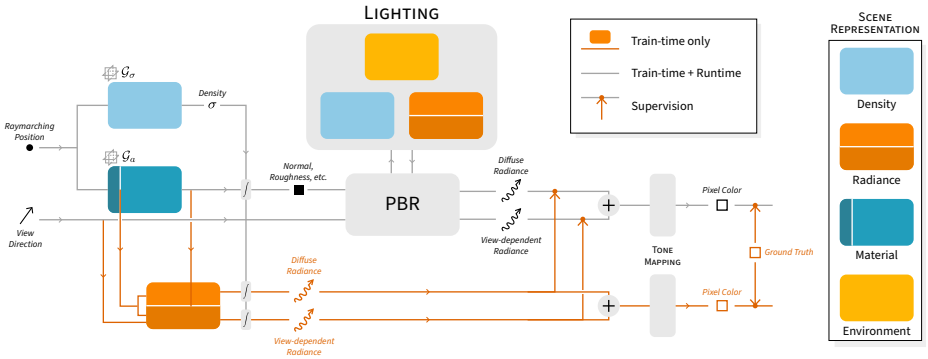

Fig. 2. Overview of our model. At each ray-marching step, we evaluate a density σ, which weights physically-based material properties and radiance information as we integrate these quantities along the marched ray. Physically-based properties are processed by our PBR fixed module to compute the final radiance. Grey boxes are fixed functions, while colored boxes are the learnable scene representation. Orange boxes and arrows are used for supervision only and dropped when evaluating with a new environment lighting. See other figures for zooms of each component.

We introduce a novel method to tackle the inverse rendering problem by leveraging efficient ray marching, neural radiance fields, as well as classical light transport knowledge. This combination allows our method to retrieve high-quality geometry and material in scenes with both highly glossy and rough surfaces. Our method takes as input a set of images of the same scene, with known camera positions under one or more unknown lighting conditions, and outputs parameters that allow to render novel views using an arbitrary new environment map.

Our method is composed of multiple **learnable components** and fixed modules that we describe in Section 4 and which constitute an end-to-end trainable architecture (Fig. 2). It includes, in particular, a physically-aware **radiance module** (Section 4.2) that bootstraps the method by retrieving coarse geometry and appearance. This module then informs a **physically-based module** which learns material and fine geometry information by leveraging a physically-aware sampling algorithm (Section 4.5). This sampling algorithm queries from

our expressive **environment map** structure (Section 4.4) based on a Laplacian Pyramid. Finally, the radiance and physically-based modules can collaborate to further disambiguate complex information in the scene (Section 5.1).

4 Architecture

We represent the optimized 3D scene through 4 learnable components (see Fig. 2.). The first 2 components are typical of NeRF-inspired methods: a **density field** encodes the coarse 3D geometry of the scene (Sec. 4.1), and a **radiance field** stores pre-integrated light information through the whole space (Sec. 4.2). The last 2 components are a **material field** (Sec. 4.5) and the **environment lighting** (Sec. 4.4): these encode quantities meant for physically based rendering.

The learnable components are connected together through differentiable fixed-function modules. The **Physically Based Rendering (PBR)** module turns physical material properties into radiance (Sec. 4.5), this requires the use of the **Lighting** module to estimate local irradiance (Sec. 4.4).

4.1 Density

The geometry of the scene is modeled as a density field. To encode this 3D scalar field, we use the TensoRF [3] representation. This enables highly efficient ray marching while being much faster to overfit than neuron-based models like NeRF [15].

The TensoRF representation decomposes 3D grids into a set of vectors and matrices. For any quantity s, we can apply bilinear interpolation on a grid $\mathcal{G}_s$ to associate to any position $x \in \mathbb{R}^3$, we note it $s_x = \mathcal{G}_s(x)$.Our 3D density tensor $\mathcal{G}_\sigma$ is thus encoded using the following decomposition $\mathcal{G}_\sigma = \sum_k \sum_{m \in XYZ} v^m_{\sigma,k} \circ M^{\tilde{m}}_{\sigma,k}$ where $v^m_{\sigma,k}, M^{\tilde{m}}_{\sigma,k}$ is the learnable decomposition. We call $\tilde{m}$ the corresponding complementary axes (e.g. $\tilde{X} = YZ$).

From this scalar field we predict the density σ_x at a given 3D location x as:

$$\sigma_x = \mathcal{G}_\sigma(x) \tag{1}$$

4.2 Physically-Inspired Radiance

Our radiance model relies on two essential ideas (Fig. 3). The decomposition of the radiance into its view dependent and independent components. The use of the directional encoding proposed by Ref-NeRF [21] to enable retrieval of correct geometry and density of specular objects. The latter enhances the former, not only we isolate the view dependent effects, but we model them in a manner that informs the predicted normals and roughness.

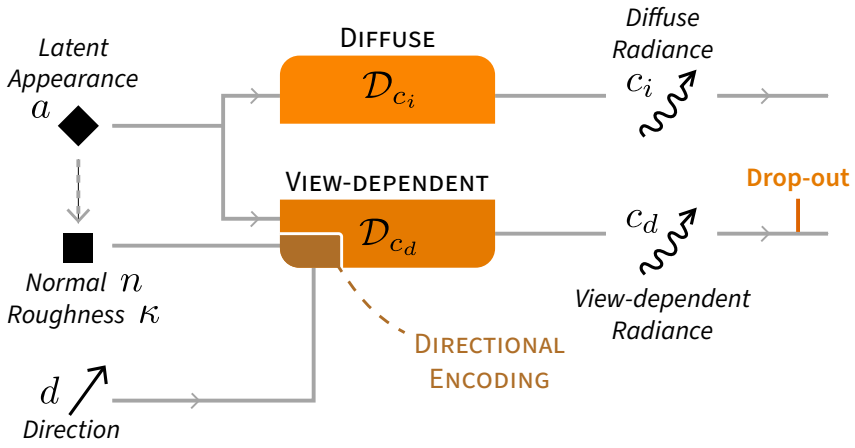

Fig. 3. Our radiance component decodes the latent appearance vector coming from the material component into view-dependent and view-independent (diffuse) terms. This is done using two isolated neural networks, only one of which receives the view direction as input. The view-dependent network is made more robust to reflections by using a directional encoding based on the prediction of the material component. A drop-out on the view-dependent term ensures that the diffuse term gets as much magnitude as possible.

Latent appearance. The radiance component inputs a latent appearance descriptor a_x shared with the material component (section 4.3). Similarly to TensoIR, we store this latent appearance information in a TensoRF field $\mathcal{G}_a = \sum_k \sum_{m \in XYZ} v^m_{a,k} \circ M^{\tilde{m}}_{a,k} \circ b^m_k$ The additional b^m_k basis vectors express the multi-channel nature of appearance (RGB). We call $a_x = \mathcal{G}_a(x)$ the latent appearance at x.

Radiance Decomposition. We introduce a decomposition that allows us to isolate the view dependent and independent visual features of a scene and thus to supervise separately the diffuse and specular terms of the PBR module (Section 5.1). As illustrated in Fig. 3, the decomposition is enforced structurally by using a different decoding network for the view-independent radiance c_i and the view-dependent radiance c_d:

$$\begin{aligned} c_i(x) &= \mathcal{D}_{c_i}(a_x) \\ c_d(x,d) &= \mathcal{D}_{c_d}(a_x, d) \end{aligned} \tag{2}$$

where x, d are coordinates and viewing direction of the current sample. Note that in general, we denote $\mathcal{D}_s$ a dense neural network and $\mathcal{D}_s(y)$ its prediction for a given input vector y. Unless otherwise stated $\mathcal{D}_s$ is a 3 layer MLP with ReLU activations. The input dimension is the sum of the different inputs and their respective Fourier features dimensions. The hidden dimensions are 128 for all layers. The output activation is a Softplus function with parameter $\beta_{\text{soft plus}} = 3$, the output dimension is the dimension of the concatenation of the predicted

quantities. This split radiance can be seen in Fig. 4. We then obtain the final radiance at training time with,

$$c(x, d) = c_i(x) + c_d(x, d) \tag{3}$$

Importantly, the neural network $\mathcal{D}_{c_d}$ contains a dropout layer during training. This is crucial for the decomposition to work (see Supp.). Contrarily to previous material prediction methods that infer albedo and specular parameters, we emphasize that we are not predicting any material properties here. Instead, through this structural choice we make the hypothesis that the radiance signal can be decomposed into view dependent and independent components. We will explore the consequences of this choice in the following.

Directional Encoding. As highlighted by Ref-NeRF [21], feeding the raw view direction d to the radiance decoding network $\mathcal{D}_{c_d}$ leads to poor learning on glossy surfaces. We use their re-parameterization, namely to work instead with the reflected vector ω_r with respect to the predicted normal n. In addition, we use their so-called Integrated Directional Encoding (IDE) to account for the aperture of the cone of reflection depending on an estimated roughness. We thus rewrite the view-dependent radiance in equation 2 as,

$$c_d(x, d) = \mathcal{D}_{c_d}\Big(a_x, \omega_r, \langle \omega_r, n_x \rangle, \mathbf{IDE}(\omega_r, \kappa_x)\Big) \tag{4}$$

where the normal n and the roughness coefficient κ are local properties predicted by our material component (Section 4.3). We refer the reader to the original Ref-NeRF paper [21] for details about the **IDE** function.

4.3 Material

We encode the physically-based material parameters in a 3D field agnostic to the current lighting condition (Fig. 5). These are used to **characterize the BSDF model** used in the PBR module (Section 4.5): a surface normal n_x, an albedo γ_x, a reflectance (specular color) $F_{0,x}$ and a roughness ρ_x parameter; and to **feed the radiance module**: κ_x, n_x.

A function mapping κ to ρ. The integrated directional encoding that we import from Ref-NeRF [21] learns its own notion of roughness κ_x as a means to provide more sensibility to viewing direction in glossy areas than in rough ones. This IDE roughness is related, but however not identical to the physically-based roughness parameter ρ_x of our BSDF model (section 5). We could decode the physically-based roughness ρ_x from the latent appearance independently from the IDE roughness κ_x, but we found empirically that having two completely separate roughness parameters may lead the model to stagnate in local-maxima (Fig. 11). We input the Fourier features of the κ_x parameter and write,

$$\rho_x = \mathcal{D}_\rho(\kappa_x)$$

The $\mathcal{D}_\rho$ differs from the introduced MLP in section 4.2, in that the hidden dimension is 10, and the output activation is a sigmoid function, i.e. it is a much smaller network.

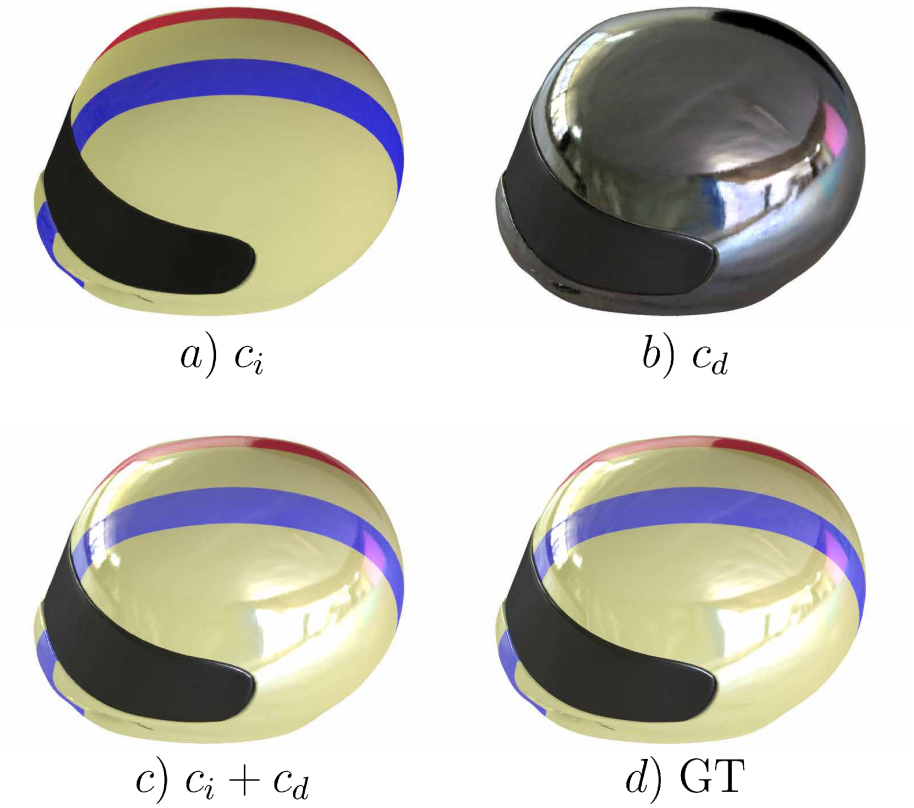

Fig. 4. Visualization of our radiance decomposition as described in Fig. 3 after overfitting on the **helmet** scene. This qualitatively corresponds to the diffuse and specular terms of a PBR BSDF model.

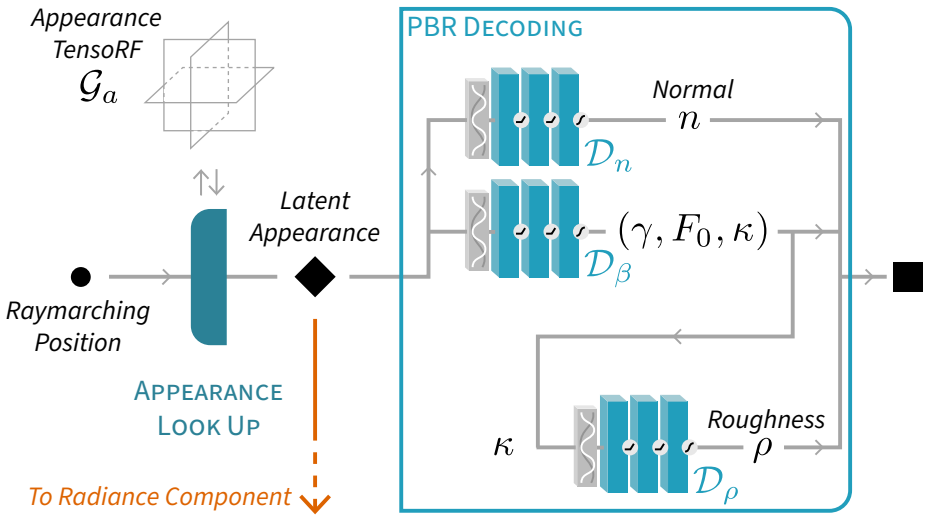

Fig. 5. The material component of our scene representation consists of a look-up in a TensoRF $\mathcal{G}_a$ followed by a simple neural network to decode the sampled latent appearance vector into physically-based material properties. This two-stage approach enables the radiance-based component to guide the definition of a latent appearance without learning a full mapping from physically-based parameters.

Leveraging appearance. We decode the remaining material quantities from the latent appearance vector a_x previously introduced in Section 4.2. More specifically:

$$\begin{aligned} n_x &= \mathcal{D}_n(a_x) \\ (\gamma_x, F_{0,x}, \kappa_x) &= \mathcal{D}_\beta(a_x) \end{aligned} \tag{5}$$

Like radiance, material properties are evaluated at each step of the raymarching, weighted by the local density σ_x, and integrated along the ray. When the accumulated density reaches a threshold, we feed the integrated properties to the PBR module. In other words, if the accumulated density is high enough, we compute the corresponding depth. We then transform this quantity into a surface point, for which we apply the PBR equation (described in section 4.5) using the integrated parameters. In the case of a "missed" ray, the parameters are not passed to the PBR module and thus are not updated. The corresponding pixel is given a default background color. This contributes to lowering the computational load. Moreover, similar to the radiance, this way of handling missed rays leads to a better posed learning objective; as opposed to forcing the network to predict the surrogate background value.

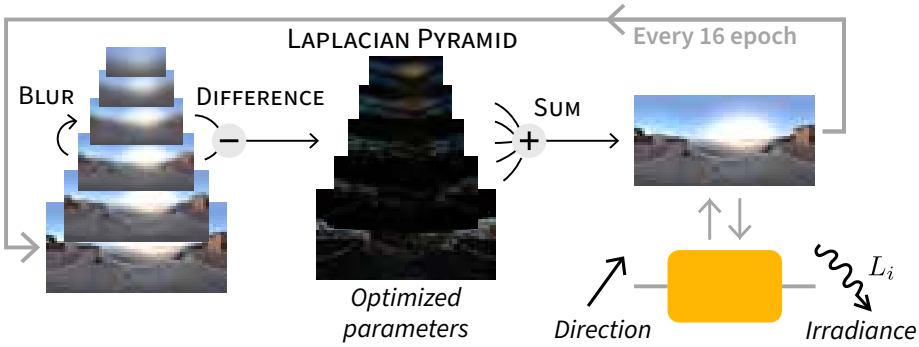

Fig. 6. In our environment lighting representation, the parameters optimized by gradient descent are the levels of a Laplacian Pyramid. This multi-scale representation better learns low frequencies than raw pixels and supports high frequencies that Spherical Harmonics cannot grasp. Every 16 epochs, we re-balance the representation to ensure that it is still a Laplacian Pyramid.

4.4 Environment Lighting

Our fourth and last learnable component encodes the retrieved environment lighting. Similarly to concurrent work [22], our method introduces a novel way to represent environment maps. However, in our case, we rely on a Laplacian Pyramid (*PoL*).

To learn high-frequency lighting details we need a high-resolution representation, but, optimizing directly the pixels of an envmap does not allow us to leverage the correlation in nearby regions. This leads to slow and noisy convergence of the lighting, as seen in Fig. 9, leading to a rough optimization landscape for the other parameters such as normals. We thus propose to optimize the levels of a Laplacian pyramid instead, allowing us to learn both low and high frequency simultaneously and to converge faster.

Given an initial envmap, we compute its Laplacian Pyramid and initialize a set of learnable parameters with the different levels. Then during optimization, at each step, we reconstruct the envmap from the parameters and bilinearly sample the reconstruction when needed. As parameters are optimized, there is no guarantee that the learned pyramid indeed represents the Laplacian Pyramid of the reconstructed signal. To enforce this, at the end of every n=16 iteration, we perform a re-projection step where we reconstruct the signal from the parameters and then compute the corresponding pyramid, reassigning the value of the parameters to these levels (Fig. 6).

PoL vs SG discussion. Previous methods such as TensoIR [7] and NeRFactor [29] represent environment maps with Spherical Gaussians (SG). Our PoL approach is better suited than SG to learn high frequency effects efficiently. While compact, SG struggle when learning high-frequency environments. Indeed, when learning high frequency environment maps from glossy scenes one needs a large number

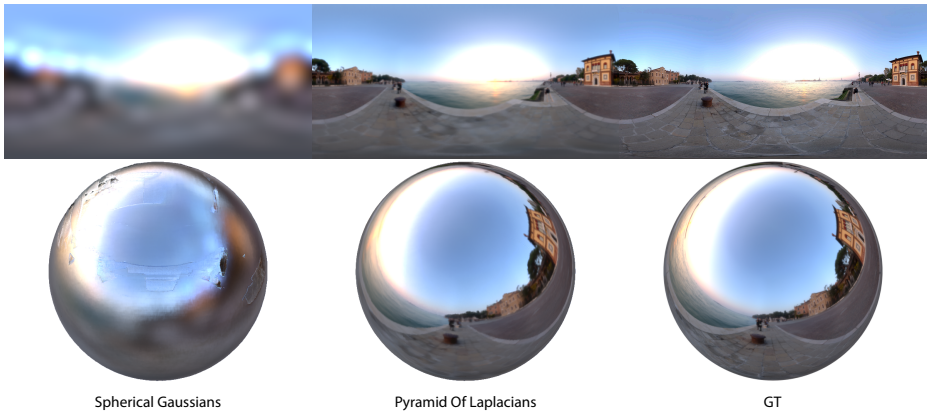

Fig. 7. Reconstructed environment map and renderings for a scene made of a simple specular sphere using different representations of the environment. SG (left) fail at extracting fine details while our proposed Laplacian Pyramid (middle) gets much closer to the ground truth (right).

of learnable parameters. In Fig. 8 we compare these approaches on the image overfitting task. We can see from this figure that in such a case our PoL approach is more memory efficient and converges faster than SG.

4.5 Physically-Based Module

Concurrently to the radiance component, the PBR module also predicts a diffuse and a view-dependent radiance. However, it does it in a physically based way, that can later be user-edited, in particular by changing the lighting condition. This module is fully differentiable, so that the error gradient may flow up to the representation of the material properties and environment light. It is summarized in Fig. 10. Below we describe the fixed function renderer used. Note that, it inputs **surface parameters**, whereas our PBR module predicts **volumetric ones**. The surface point evaluated is predicted using the estimated depth, and the physically-based parameters described in equation 5 are aggregated along the associated marched ray to obtain the surface parameters.

We settled for such an approach to transfer the learning flexibility of ray-marching, used by our radiance module, to the PBR module, which intends to condense light-geometry interaction to surface by fitting a surfacic BSDF model. This also mitigates the cost of indirect lighting evaluation. The accumulation-based evaluation of normals and roughness ensures graceful degradation, either when this hypothesis is wrong or while the model did not converge yet. Consistency along a ray is progressively ensured by the alignment supervision, and when the scene is indeed surfacic the density that weights accumulation is eventually null anywhere but on surfaces.

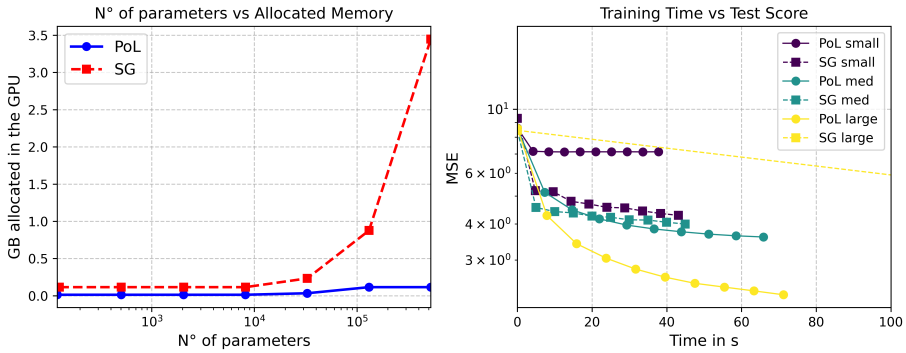

Fig. 8. Experiment: Fitting the PoL and SG approaches on a given ground truth image. This simplified task gives us insight on how these methods compare. Left: we have a plot that shows the variation in allocated memory at train time as the number of parameters of each method increases. Right: Elapsed training time plotted against the test reconstruction score. Our PoL approach converges extremely fast, and larger models lead to better test results. A SG approach with a large number of lobes is challenging to train.

Fig. 9. Directly optimizing the pixels of the envmap (top - corresponding to a single level PoL) leads to a noisier envmap with boundaries artifacts (insets). Using a 6-level PoL (bottom) provides a smoother estimate.

Rendering. The physically-based radiance c_{PB} is computed based on the rendering equation:

$$c_{PB}(\hat{x}, \omega_o) = \int_{\Omega} L_i(\hat{x}, \omega_i) f_r(\omega_o, \omega_i; \beta) \langle \omega_i, n \rangle_+ \, d\omega_i \tag{6}$$

where $\hat{x}$ is the surface point, $\omega_o = -d$ is the viewing direction, $L_i(\hat{x}, \omega_i)$ is the incident illumination coming from a direction ω_i, $\beta := (\gamma, F_0, \rho)$ are the material properties and n is the normal at $\hat{x}$.

The BRDF f_r can be split into diffuse and specular (view-dependent) terms:

$$f_r(\omega_o, \omega_i; \beta) = f_{\text{diffuse}}(\gamma) + f_{\text{specular}}(\omega_o, \omega_i; \beta) \tag{7}$$

We integrate these terms separately as c_{PB}^{dif} and c_{PB}^{spec}, to be able to supervise them using respectively outputs c_i and c_d of the radiance component. In practice, our spatially varying BRDF model is based on the Torrance–Sparrow model with a normal distribution function based on the Beckmann–Spizzichino model [2] (see Supp.).

Irradiance. For each light ray sampled by the MIS scheme (see Supp.), we evaluate the light intensity L_i coming from that direction. We leverage the efficient ray marching procedure of TensoRF [3] to query the incident illumination using the radiance module if the ray hits the scene, or using our environment light component otherwise.

When later evaluating the scene on a new unseen light condition for which the radiance component has no information, this radiance component is replaced by a recursive call to the PBR module, just like in a traditional ray tracer.

5 Optimization scheme

Our overall optimization procedure uses a typical machine learning approach: we optimize our learnable components with a gradient descent using the common AdamW optimizer [12] and evaluate gradients using the automatic differentiation of PyTorch. This section details some mechanisms we used to improve the convergence of our model.

5.1 Supervision

Overall the loss we optimize is a weighted sum of the following terms:

- l_{RF}, l_{PB} the photometric (l2) losses produced by the radiance and PB modules respectively.
- l_{diffuse},l_{specular} which we call our supervision losses on the decomposition and introduce below.
- $l_n = \sum_{w_j} ||n_j - n_{\sigma,j}||_2^2$, the normal alignment loss introduced by Ref-NeRF. A loss term penalizing back-facing normals is used in addition.

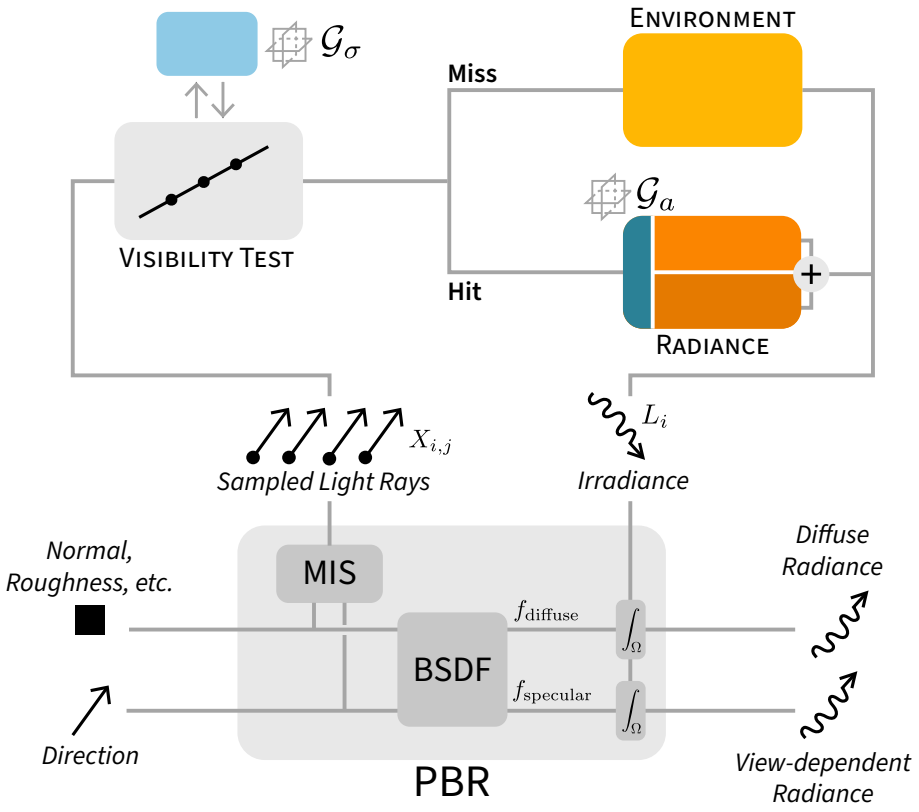

Fig. 10. Our Physically-Based Rendering module uses Multiple Importance Sampling to estimate the incoming light at the shaded point. For each sampled light ray, we use either the environment or the radiance component depending on a ray marching through the density grid. For novel-light synthesis, the radiance component is replaced by a recursive call to the PBR module.

– l_β to ensure local smoothness loss on the different PB parameters, a Total Variation (TV) loss and l_1 regularization on tensor factors from [7, 3]

The radiance loss l_{RF} drives the training procedure so it is the loss with the highest weight. While most of these terms were already used by TensoIR, we introduce:

$$l_{\text{diffuse}} = \|c_{PB}^{\text{dif}}(\hat{x}) - c_i(\hat{x})\|^2 \text{ and,}$$
$$l_{\text{specular}} = \|c_{PB}^{\text{spec}}(\hat{x}, d) - c_d(\hat{x}, d)\|^2$$

Supervising the diffuse and specular terms of the BSDF independently helps the disambiguation of intricate visual information from the input images. Although the inverse rendering problem is inherently ambiguous, this physically motivated prior results in higher quality retrieved parameters which in return improves the relighting performance.

5.2 Radiance warm-up

Our physically-aware radiance module is capable of learning notions of normals and roughness by itself, following Ref-NeRF. Moreover, our method heavily relies on a radiance module that has “understood" the coarse geometry of the scene before starting the PBR procedure. Indeed, it is the radiance module that manages to process strong highlights and complex geometry in an efficient way. We therefore warm-up our radiance module by training it for 30k iterations (∼1h) before enabling the PBR module.

To obtain optimal results we let our PBR module then run for another 70k iterations (∼20h) on an NVIDIA Tesla T4 GPU. We have not sought to optimize this parameter and runtime in this paper. We empirically note that the time and number of iterations needed to achieve the best result in each scene significantly varies (∼1h-20h).

6 Experiments

Our method is capable of processing scenes comprised of both glossy and diffuse elements. We thus compare it to two state-of-the-art **similar** inverse rendering papers, TensoIR [7] which performs best on **diffuse** scenes and NMF [13] for **glossy** ones. We use Fig. 12 to highlight the ability of our method to tackle both types of scenes. We choose in this qualitative comparison objects for which our method outperforms the aforementioned papers.

Additionally, we perform ablation studies that provide a quantitative justification for our main contributions. Our method is tested on novel-view synthesis (NVS) and relighting tasks on two synthetic datasets: the TensoIR Synthetic dataset from [7], and the Shiny Blender dataset from Ref-NeRF [21]. For any other shapes we use the pipeline introduced by NeRFactor [29]. Since these are synthetic datasets the camera information is directly extracted from blender. To

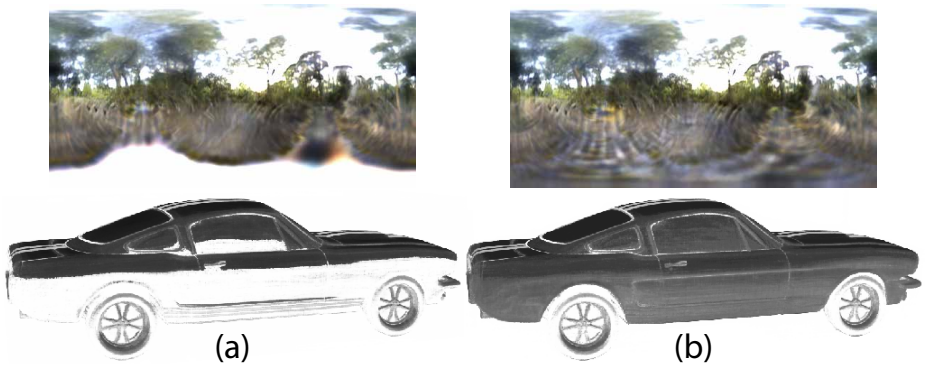

Fig. 11. Visualizing ρ with (a) separate roughness parameters, and (b) using our $\kappa \mapsto \rho$ map. The learnt environment map by each model is also visualized. We see that bad learning of the (a) model leads to loss of information in the environment map.

evaluate performance on these two tasks we employ the standard metrics PSNR, SSIM [23], and LPIPS [28]. The **quality of our reconstructed normal** is one of the main features of our method, we measure it with the Mean Angular Error (MAE°).

Comparison with NMF. Our method is able to retrieve normals of much better quality than NMF, as illustrated in Fig. 13 and confirmed numerically with the MAE error in Table 6. This table also reports that our model does not match the quantitative similarity scores performance presented by NMF on NVS task. Our model under performing on the NVS task is likely linked to the use of a neural components in NMF's BSDF model. This is not a component that is pre-trained on different materials, rather trained from scratch for each scene. This helps the overfitting of the scene, which is helpful in reconstruction. However, it does not generalize well to other light environments.

Nevertheless one can appreciate in Fig. 13 that our novel views feature more consistent reflections. Indeed, we remark this qualitative improvement in multiple scenes, and the benefits of our good normal extraction becomes clear when it comes to relighting tasks: Table 6 shows that we outperform NMF on the relighting task quantitatively on the *shiny blender* dataset that the NMF paper focuses on. Figure 12 shows visual examples of such relighting.

When comparing results and figures one must note that NMF uses HDR input files to train and test their model. We rather use LDR images as they are a more commonly available in practical scenarios. Moreover, we would like to highlight that the qualitative examples for NMF were directly provided by the authors of the paper. This is because, at the time of our experiments, we were unable to retrieve high quality results with the publicly available code.

Comparison with TensoIR. We ran TensoIR [7] on the same *shiny blender* dataset as the NMF comparisons, and as we see on the bottom row of Table 6

our method outperforms TensoIR in all metrics on NVS and normal extraction tasks. As a matter of fact, TensoIR does not perform as well on shiny scenes in general. Figure 14 shows that even on the more diffuse scenes that TensoIR targets, we **maintain** PBR quality while slightly improving on the retrieved normals. In this figure we can see however, that the MIS algorithm comes with some undesired noise. Indeed, whereas using fixed light sampling limits the rendering of shiny materials, it produces less noisy results for diffuse objects than MIS for the same number of samples.

TensoIR uses the learned radiance as a proxy for indirect lighting during training. This however, cannot be done during testing on different lighting conditions. Thus, the authors decided to omit indirect lighting during their evaluation process. Our framework allows for evaluation of indirect lighting, thanks to a **slightly** more sophisticated renderer. This is what we use for our teaser (figure 1), which explains why we can see reflections on the relit scene. For a proper comparison on the relighting task we restrict our comparison to the TensoIRSynthetic dataset and do not compute indirect lighting. Table 6 reports that our method is slightly over performed by TensoIR, but manages to **maintain** high quality results by achieving better results than other well-established methods such as NeRFactor [29] and InvRender [30]. Given that important parameters, such as the normals, are better predicted by our method we can attribute the slight difference in performance to our **noisier** rendering pipeline, due to our use of MIS to sample light directions as previously discussed.

Ablations. Table 6 shows results on a novel view synthesis task for different ablations of our pipeline. Each ablation removes one of the building blocks of the method. First, "w/ separate ρ" learns the PBR ρ_x directly as an output of $\mathcal{D}_\beta$ in equation 5, independently from the IDE roughness κ_x, "w/o Decomposition" replaces our decomposition introduced via equation 2 by the same radiance map used by TensoRF. Finally, "w/o Supervision" omits the losses introduces in section 5.1, l_{diffuse},l_{specular}.

We can see in Table 6 our ablation results for the relighting tasks. Although "Ours separate ρ" allows for slightly better normal reconstruction, we see ultimately the benefit of our $\kappa \mapsto \rho$ map in relighting scores. Indeed, this mapping is important to better retrieve the roughness in ambiguous scenes. In figure 11, we can see how utilizing our $\kappa \mapsto \rho$ mapping allows to leverage the notion of roughness learned by the radiance to avoid losing information. The "w/o Supervision" seems to yield quite similar results to our method. However, we can see in the detailed tables presented in our supplemental section, that supervision helps in scenes where the decomposition is the cleanest (figure 4) such as helmet or toaster. In diffuse scenes our method without supervision performs better. One could alleviate this by setting a smaller weight for the supervision. We decided to use common weights among our tests to present a fair comparison.

Our Laplacian Pyramid model for environment lighting is compared to the mixture of SG used by TensoIR in section 4.4. Although very flexible for learning rough lighting, it eventually fails at grasping the fine details of the environment. This is not a problem for rough objects, but as shown on the toy sphere exam-

ple, it makes it impossible to properly learn very glossy materials and leads to artifacts in the albedo and roughness. Lastly, Fig 9 highlights that the multi-scale approach of the Laplacian Pyramid benefits to the final quality of the environment reconstruction.

Table 1. Quantitative comparison on the shiny blender test. The PSNR, SSIM and LPIPS scores measure the similarity between novel view synthesis and ground truth under the same light condition. The MAE score characterizes the reconstruction of the geometry, which is independent from any lighting. We highlight the **best** and the second best scores.

	PSNR ↑	**SSIM**↑	**LPIPS** ↓	**MAE**↓
Ours	31.635	0.941	0.098	2.262
w/ separate ρ	32.009	0.945	0.098	**2.197**
w/o Decomposition	31.235	0.936	0.108	2.809
w/o Supervision	32.289	0.947	0.098	2.260
TensoIR	31.296	0.939	0.089	4.390
NMF	**33.599**	**0.958**	**0.046**	3.659

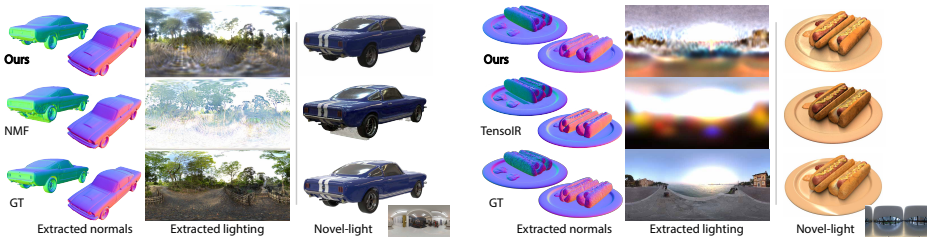

Fig. 12. Comparison with TensoIR [7] and NMF [13] showing that our method better retrieves PBR parameters like normals and environment lighting, and thus leads to better relighting, especially on scenes featuring glossy surfaces.

7 Discussion

We have introduced a novel and powerful approach to tackle the ambiguous problem of inverse rendering. Using a set of input images with their respective camera information we can generate the geometry, environment illumination, and material properties of the scene. We leverage both volumetric rendering through the use of our radiance module, and physically-based rendering.

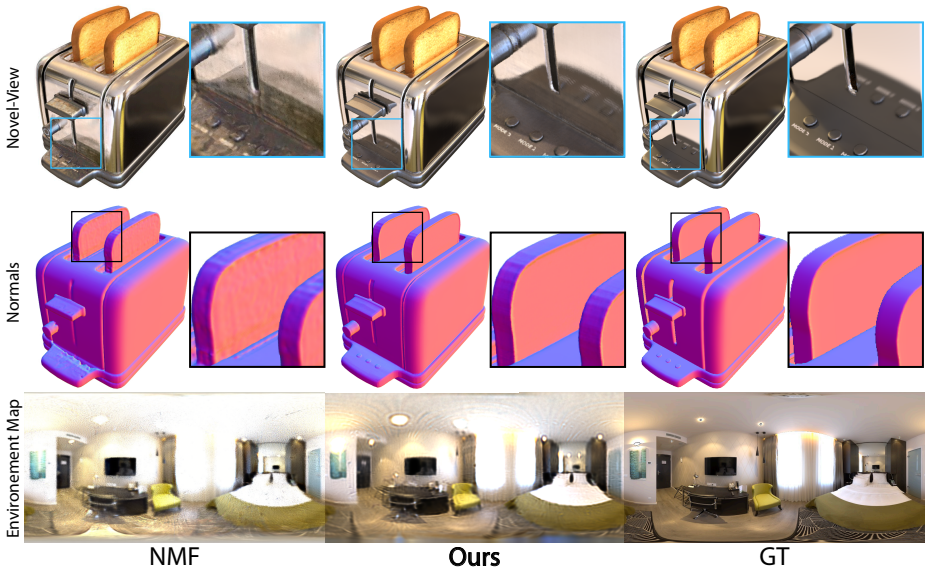

Fig. 13. Comparison with NMF [13] showing that our method outperforms the state of the art in novel-view synthesis for glossy surfaces. In particular, we avoid the ghosting artifact seen in the self-reflections of NMF.

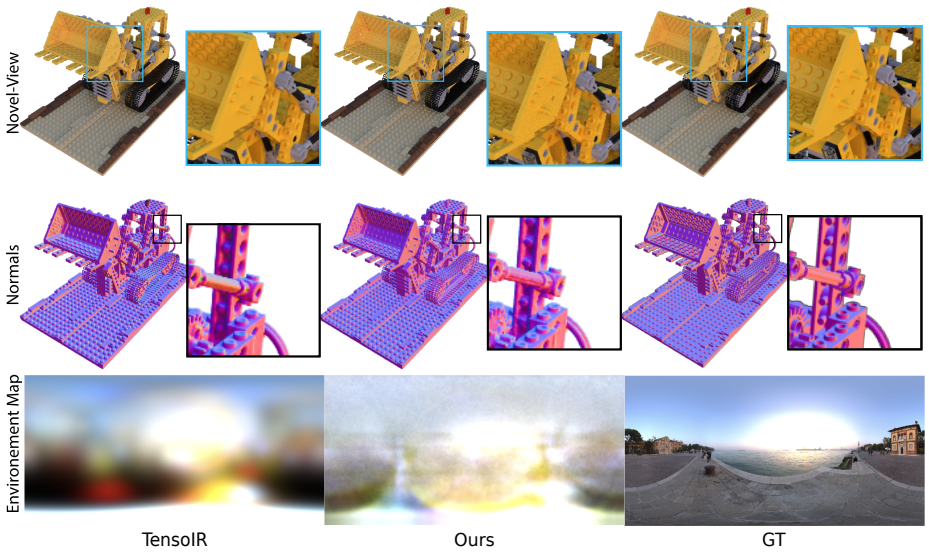

Fig. 14. Comparison with TensoIR [7] showing that our method matches the state of the art in novel-view synthesis for diffuse surfaces. Our rendering is slightly noisier due to our importance sampling approach.

Table 2. Quantitative comparison of the similarity between relighted scenes and ground truth on the shiny blender dataset. We highlight the **best** and the second best scores.

	PSNR ↑	**SSIM↑**	**LPIPS ↓**
Ours	**25.838**	**0.925**	**0.101**
NMF	25.502	0.916	0.113
NVDiffRec	20.686	0.8312	0.191
NVDiffRecMC	22.196	0.874	0.2158

Table 3. Quantitative comparison ablation of relighting task. We highlight the **best** and the second best scores.

	PSNR ↑	**SSIM↑**	**LPIPS ↓**
Ours	**25.261**	0.924	0.096
w/ separate ρ	25.113	0.921	0.097
w/o Decomp	24.886	0.915	0.105
w/o Supervision	25.179	**0.926**	**0.091**

Table 4. Quantitative comparison between relighted scenes on the TensoIR Synthetic dataset. We highlight the **best** and the second best scores.

	PSNR ↑	**SSIM↑**	**LPIPS ↓**
Ours	28.144	0.929	0.085
TensoIR	**28.58**	**0.944**	**0.081**
NeRFactor	23.383	0.908	0.131
InvRender	23.973	0.901	0.101

7.1 Properties and insights

We have shown in our quantitative and qualitative tests that our method is suited for a wide a range of effects. It can capture scenes composed of complex geometry (Fig. 14) and it excels at tackling glossy surfaces (Fig. 13).

In contrast to state-of-the-art methods such as NeRO [11], which advocate for processing information through separate geometry and material stages, our method tackles the problem with a single-stage, end-to-end optimizable architecture. One of NeRO's limitation according to the authors is the failure of retrieving subtle geometrical details. Even though our method is not put to the test on real-life data, the quality of the normals we achieve on our tests seem to indicate that our model does not have this limitation.

Our contributions advocate for similar methods. That is, end-to-end optimizable radiance guided approaches to inverse rendering. We have showed on this paper that NeRFs can provide good coarse features. Indeed, we achieve state-of-the-art extraction of normals. This is a crucial step in scene understanding, since given a neat normal, the learning of the other parameters is better constrained. Furthermore, NeRFs can help to disambiguate the inverse rendering problem, our radiance decomposition leads to an increase in performance in all our tests, and using the roughness predicted by the radiance can help to escape local extrema. As opposed to methods that focus on rendering predicted PB parameters (NMF [13], NeRO[11]) we show that dual rendering approaches can help to better condition the ill posed problem of inverse rendering.

7.2 Limitations

Our model is not without limitations; each of its components may be limiting in some situation. If the radiance component cannot "understand" the scene enough to initiate the extraction, the PBR module cannot help it getting out of strong local minima. Like the methods we compare to, we focused our tests on surfacic scenes, with no semi-transmissive volumes. If the scene contains light scattering effects that our PBR module cannot replicate (e.g., subsurface scattering, iridescence, etc.), it will only try to fit its BSDF model, which may notably mess up with normal extraction. Our model will not perform optimally. Moreover, strong inter-reflections are very hard to properly extract, and our method fails when the far lighting assumption is not met (See Supp.).

Even if we could perfectly reconstruct the geometry of the scene, the environment light and material properties are only retrieved up to a multiplicative ambiguous parameter: multiplying the lighting by a fixed factor can be balanced by globally reducing the albedo and reflectance. More generally, it is difficult to find common ground for comparison in the space of physically-based parameters, as they rely on different BSDF models, some of which are even partially learned [13]. This is why we focused our efforts on normals. Ultimately, the choice of BSDF model depends on downstream use of the extracted 3D. The quantitative evaluation of the environment reconstruction is also a question: errors in areas that do not contribute to the rendered images, or only contribute to rough surfaces should not be penalized in the same way as sharp reflections.

7.3 Future Work

Each of our components can be individually improved. Most insights of our system are not specific to the TensoRF model for positional encoding, so other learnable representations could be used. Our environment light component could be augmented in order to support not only far directional light, but also near distance light sources, like out-of-frustum surfaces, which we would typically meet when applying our system to non-synthetic images. We could explore the behavior of our approach in presence of transmissive surfaces, leading to multiple calls to the PBR module per camera ray, or its interaction with rasterization-based inverse rendering like 3D Gaussian Splatting paper [8]. Lastly, our method still has a lot of room to be optimized. There exist optimized NeRF libraries from which our method could benefit from [10].

Acknowledgments. We wish to warmly thank the authors of TensoIR [7] and NMF [13] for providing details and results beyond what was available in the original publications.

References

1. Azinovic, D., Li, T.M., Kaplanyan, A., Nießner, M.: Inverse path tracing for joint material and lighting estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2447–2456 (2019)
2. Beckmann, P.: Spizzichino, the scattering of electromagnetic waves from rough surfaces (1963)
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
4. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. p. 43–54. SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237170.237200, https://doi.org/10.1145/237170.237200
5. Hasselgren, J., Munkberg, J., Lehtinen, J., Aittala, M., Laine, S.: Appearance-driven automatic 3d model simplification. In: EGSR (DL). pp. 85–97 (2021)
6. Jambon, C., Kerbl, B., Kopanas, G., Diolatzis, S., Leimkühler, T., Drettakis, G.: Nerfshop: Interactive editing of neural radiance fields". Proceedings of the ACM on Computer Graphics and Interactive Techniques **6**(1) (May 2023), https://repo-sam.inria.fr/fungraph/nerfshop/
7. Jin, H., Liu, I., Xu, P., Zhang, X., Han, S., Bi, S., Zhou, X., Xu, Z., Su, H.: Tensoir: Tensorial inverse rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 165–174 (June 2023)
8. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
9. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. p. 31–42. SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237170.237199, https://doi.org/10.1145/237170.237199
10. Li, R., Tancik, M., Kanazawa, A.: Nerfacc: A general nerf acceleration toolbox. arXiv preprint arXiv:2210.04847 (2022)

11. Liu, Y., Wang, P., Lin, C., Long, X., Wang, J., Liu, L., Komura, T., Wang, W.: Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. arXiv preprint arXiv:2305.17398 (2023)
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
13. Mai, A., Verbin, D., Kuester, F., Fridovich-Keil, S.: Neural microfacet fields for inverse rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 408–418 (October 2023)
14. Martel, J.N., Lindell, D.B., Lin, C.Z., Chan, E.R., Monteiro, M., Wetzstein, G.: Acorn: Adaptive coordinate networks for neural representation. ACM Trans. Graph. (SIGGRAPH) (2021)
15. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
16. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127
17. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in neural information processing systems **33**, 7462–7473 (2020)
18. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021)
19. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 7537–7547. Curran Associates, Inc. (2020)
20. Toschi, M., De Matteo, R., Spezialetti, R., De Gregorio, D., Di Stefano, L., Salti, S.: Relight my nerf: A dataset for novel view synthesis and relighting of real world objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20762–20772 (June 2023)
21. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022)
22. Verbin, D., Mildenhall, B., Hedman, P., Barron, J.T., Zickler, T., Srinivasan, P.P.: Eclipse: Disambiguating illumination and materials using unintended shadows. arXiv (2023)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
24. Xu, Y., Zoss, G., Chandran, P., Gross, M., Bradley, D., Gotardo, P.: Renerf: Relightable neural radiance fields with nearfield lighting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22581–22591 (October 2023)
25. Yifan, W., Serena, F., Wu, S., Öztireli, C., Sorkine-Hornung, O.: Differentiable surface splatting for point-based geometry processing. ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA) **38**(6) (2019)

26. Yuan, Y.J., Sun, Y.T., Lai, Y.K., Ma, Y., Jia, R., Gao, L.: Nerf-editing: geometry editing of neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18353–18364 (2022)
27. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5453–5462 (2021)
28. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
29. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. ACM Transactions on Graphics (ToG) **40**(6), 1–18 (2021)
30. Zhang, Y., Sun, J., He, X., Fu, H., Jia, R., Zhou, X.: Modeling indirect illumination for inverse rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18643–18652 (2022)
31. Zhou, X., Kalantari, N.K.: Adversarial single-image svbrdf estimation with hybrid training. In: Computer Graphics Forum. vol. 40, pp. 315–325. Wiley Online Library (2021)